CLAIMS

What is claimed is:

1. A method for matrix transposition, the method comprising:
rotating in a vector register a first row of a matrix to generate a first row of elements;
writing simultaneously into a plurality of look up units the first row of elements indexed by a first row of indices in a vector register;
looking up simultaneously from the plurality of look up units a second row of elements indexed by a second row of indices in a vector register; and
rotating in a vector register the second row of elements to generate a third row of elements.

2. A method as in claim 1 wherein each element of the matrix comprises a plurality of bit segments, each of which is written into an entry of a different unit of the plurality of look up units.

3. A method as in claim 1 wherein the plurality of look up units are configured into a plurality of look up tables in response to receiving an instruction for looking up a row of elements.

4. A method as in claim 1 further comprising:
concurrently rotating in a vector register a second row of matrix to generate a forth row of elements while writing the first row of elements.

5. A method as in claim 4 wherein a row that needs no rotation is written into look up units before other rows are written into the look up units.

6. A method as in claim 4 further comprising:
concurrently computing a third row of indices using the first row of indices while writing the first row of elements.

7. A method as in claim 6 further comprising:
concurrently loading a row of the matrix from memory into a vector register while writing the first row of elements.

8. A method as in claim 6 wherein:
the first row of indices are a first constant;
the third row of indices are a second constant; and
the first and second constants differ by one.

9. A method as in claim 6 wherein the third row of indices is a result of a rotation of the first row of indices.

10. A method as in claim 1 further comprising:
concurrently rotating in a vector register a fifth row of elements to generate a forth row of elements while looking up the second row of elements.

11. A method as in claim 10 wherein a row of elements that needs no rotation is looked up from the plurality of look up units after other rows are looked up from the plurality of look up units.

12. A machine readable media containing executable computer program instructions which when executed by a digital processing system cause said system to perform a method for matrix transposition, the method comprising:
rotating in a vector register a first row of a matrix to generate a first row of elements;
writing simultaneously into a plurality of look up units the first row of elements indexed by a first row of indices in a vector register;
looking up simultaneously from the plurality of look up units a second row of elements indexed by a second row of indices in a vector register; and
rotating in a vector register the second row of elements to generate a third row of elements.

13. A media as in claim 12 wherein each element of the matrix comprises a

plurality of bit segments, each of which is written into an entry of a different unit of the plurality of look up units.

14. A media as in claim 12 wherein the plurality of look up units are configured into a plurality of look up tables in response to receiving an instruction for looking up a row of elements.

15. A media as in claim 12 wherein the method further comprises:
concurrently rotating in a vector register a second row of matrix to generate a forth row of elements while writing the first row of elements.

16. A media as in claim 15 wherein a row that needs no rotation is written into look up units before other rows are written into the look up units.

17. A media as in claim 15 wherein the method further comprises:
concurrently computing a third row of indices using the first row of indices while writing the first row of elements.

18. A media as in claim 17 wherein the method further comprises:
concurrently loading a row of the matrix from memory into a vector register while writing the first row of elements.

19. A media as in claim 17 wherein:

the first row of indices are a first constant;

the third row of indices are a second constant; and

the first and second constants differ by one.

20. A media as in claim 17 wherein the third row of indices is a result of a rotation of the first row of indices.

21. A media as in claim 12 wherein the method further comprises:

concurrently rotating in a vector register a fifth row of elements to generate a forth row of elements while looking up the second row of elements.

22. A media as in claim 21 wherein a row of elements that needs no rotation is looked up from the plurality of look up units after other rows are looked up from the plurality of look up units.

23. A processing system for matrix transposition, the system comprising:

means for rotating in a vector register a first row of a matrix to generate a first row of elements;

means for writing simultaneously into a plurality of look up units the first row of elements indexed by a first row of indices in a vector register;

means for looking up simultaneously from the plurality of look up units a second row of elements indexed by a second row of indices in a vector register; and

means for rotating in a vector register the second row of elements to generate a third row of elements.

24. A processing system as in claim 23 wherein each element of the matrix comprises a plurality of bit segments, each of which is written into an entry of a different unit of the plurality of look up units.

25. A processing system as in claim 23 wherein the plurality of look up units are configured into a plurality of look up tables in response to receiving an instruction for looking up a row of elements.

26. A processing system as in claim 23 further comprising:
means for concurrently rotating in a vector register a second row of matrix to generate a forth row of elements while writing the first row of elements.

27. A processing system as in claim 26 wherein a row that needs no rotation is written into look up units before other rows are written into the look up units.

28. A processing system as in claim 26 further comprising:
means for concurrently computing a third row of indices using the first row of indices while writing the first row of elements.

29. A processing system as in claim 28 further comprising:
means for concurrently loading a row of the matrix from memory into a vector register while writing the first row of elements.

30. A processing system as in claim 28 wherein:
the first row of indices are a first constant;
the third row of indices are a second constant; and
the first and second constants differ by one.

31. A processing system as in claim 28 wherein the third row of indices is a result of a rotation of the first row of indices.

32. A processing system as in claim 23 further comprising:
means for concurrently rotating in a vector register a fifth row of elements to generate a forth row of elements while looking up the second row of elements.

33. A processing system as in claim 32 wherein a row of elements that needs no rotation is looked up from the plurality of look up units after other rows are looked up from the plurality of look up units.

34. A processing system for matrix transposition, the system comprising:

a vector register file comprising a plurality of vector registers;

a vector processing unit coupled to the vector register file, the vector processing unit comprising a vector look up unit, the vector look up unit comprising a plurality of look up units adapted to look up a vector of data items simultaneously, the vector processing unit:

rotating in a vector register in the vector register file a first row of a matrix to generate a first row of elements;

writing simultaneously into the plurality of look up units the first row of elements indexed by a first row of indices in a vector register in the register file;

looking up simultaneously from the plurality of look up units a second row of elements indexed by a second row of indices in a vector register in the register file; and

rotating in a vector register in the vector register file the second row of elements to generate a third row of elements.

35. A processing system as in claim 34 wherein each element of the matrix comprises a plurality of bit segments, each of which is written into an entry of a different unit of the plurality of look up units.

36. A processing system as in claim 34 wherein the plurality of look up units are configured into a plurality of look up tables in response to receiving an instruction for looking up a row of elements.

37. A processing system as in claim 34 wherein the vector processing unit concurrently rotates in a vector register a second row of matrix to generate a forth row of elements while writing the first row of elements.

38. A processing system as in claim 37 wherein a row that needs no rotation is written into look up units before other rows are written into the look up units.

39. A processing system as in claim 37 wherein the vector processing unit concurrently computes a third row of indices using the first row of indices while writing the first row of elements.

40. A processing system as in claim 39 wherein the vector processing unit concurrently loads a row of the matrix from memory into a vector register while writing the first row of elements.

41. A processing system as in claim 39 wherein:
the first row of indices are a first constant;
the third row of indices are a second constant; and
the first and second constants differ by one.

42. A processing system as in claim 39 wherein the third row of indices is a result of a rotation of the first row of indices.

43. A processing system as in claim 34 wherein the vector processing unit concurrently rotates in a vector register a fifth row of elements to generate a forth row of elements while looking up the second row of elements.

44. A processing system as in claim 43 wherein a row of elements that needs no rotation is looked up from the plurality of look up units after other rows are looked up from the plurality of look up units.